REMARKS/ARGUMENTS

Claims 2-55 are pending in this application and are rejected. Claims 8, 20, 37, and 44 are currently amended, and such amendments are fully supported by the specification and drawings, at least at page 16, lines 6-19. For at least the reasons set forth below, Applicants assert that all claims are in condition for allowance.

Rejection under 35 U.S.C. § 103

Claims 2-55 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Dillingham U.S. Patent No. 6,327,608 in view of Wolf et al. U.S. Patent No. 5,818,447. As set forth in more detail below, there is no suggestion or motivation to modify or combine the references and the references fail to teach or suggest all the claim limitations as required by MPEP § 2143. Therefore, the rejection is unsupported by the art, and Applicants respectfully request that the rejection be withdrawn.

(a) *There is No Suggestion or Motivation to Combine the References*

Even assuming *arguendo* that the cited references can be successfully combined, there is no suggestion or motivation in the record to do so. Accordingly, a *prima facie* case of obviousness has not been established. *See* MPEP § 2143.01 ("The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination.").

The rejection suggests that it would have been obvious "to modify the UI form definition taught by Dillingham to include the server-based application of Wolf et al., in order to access to utilizes [sic] native client user interface features to display data received from a server as taught by Wolf et al." OA dated November 17, 2004, p. 4. Applicants respectfully disagree with this assessment. *Dillingham* describes a system for remotely browsing and administering physical file directories. In contrast, *Wolf* describes a system for locally viewing and editing email messages on computer 10.

It is implausible that one of ordinary skill in the art, when working with the system for remote file administration of *Dillingham*, would turn to a system for editing email messages in order to use a native client user interface to display server data. It is unlikely that one skilled in the art would use *Dillingham* in any effort to browse and administer remote files with a native client user interface because such client interfaces already existed, for example Windows Explorer. The foundation for *Dillingham* is file browsing and administration that is performed through a web browser receiving remote HTML and XML instructions. One skilled in the art would not attempt to modify such a configuration with a system for locally editing email messages.

Moreover, the rejection only asserts that modifying the user interface of *Dillingham* with *Wolf* would have been obvious if one of ordinary skill in the art had "the teachings of Dillingham et al. and Wolf et al. before them..." OA dated November 17, 2004, p. 4. Even if this statement is true, it begs the question of whether one of ordinary skill in the art would have both references before him or her. As shown above, there is no evidence in the record that one of ordinary skill in the art would seek out the *Wolf* reference in order to modify the system of *Dillingham*.

(b) *The References Fail to Teach or Suggest an Offline Action*

Claim 8 recites, "receiving a command from said client device, said command being indicative of an offline action performed by said client device." As previously demonstrated, the cited references fail to teach or suggest this limitation. Examiner cites to *Dillingham* at Col. 2, lines 27-64 in support of the rejection of this limitation. However, this passage fails to teach or suggest the receipt of a command indicative of an offline action as claimed. Specifically, *Dillingham* discloses a file administration application that provides remote network access to a server-based file system via a client computer. The *Dillingham* system employs a client PC connected to the file server via the Internet. In this regard, one of the requirements of the *Dillingham* system is the PC-to-server connection. Indeed, *Dillingham* states that "[p]rior to gaining access to the server file system, the client first establishes a secure connection with the server." Col. 6, lines 31-32. *Dillingham* also states that "[t]he remote file administration architecture operates within the context of this secured environment. Accordingly, the commands described below...are all securely exchanged over the Internet..." Col. 6, lines 48-52.

In response to Applicants' earlier arguments on this point, Examiner cites a specific passage from *Dillingham*, col. 2, lines 30-31: "Applicant's attention is directed to the lines 'The UI might be stored locally at the client, or downloaded on demand from the server.'" OA dated April 7, 2004, p. 7. Applicants have carefully reviewed this disclosure, but it still fails to teach or suggest the receipt of a command indicative of an offline action. Nowhere does *Dillingham* describe the relationship between the client PC and the file server as "offline." Indeed, to the contrary, everything in the *Dillingham* reference suggests a <u>online</u> relationship between the client PC and the file server.

Additionally, merely because "the UI [of *Dillingham*] might be stored locally at the client, or downloaded on demand from the server" does not suggest that the server receives a command indicative of an offline action. Under either of these two scenarios, the UI may communicate to the server only <u>online actions</u>, and there is no suggestion in *Dillingham* to the contrary. Nowhere does the *Dillingham* reference, nor the other art of record, teach or suggest modifying the *Dillingham* reference to include the receipt of a command indicative of an offline action.

(c) *The References Fail to Teach or Suggest the Claimed Configuration of UI Form Definitions and Source Data Items*

Claims 8 and 20 recite (1) a UI form definition that corresponds to or is based on a particular client device's platform or the client device's capabilities; (2) the UI form definition is stored at or transmitted from a UI server; (3) a cached copy of the UI definition also is saved on the client device; and (4) transmitting or retrieving source data items from the UI server to populate the UI form on the client device. Claims 37 and 40 similarly recite items (1), (3), and (4), and transmitting a UI form identifier to the client device. The rejection asserts that these limitations are taught or suggested by *Dillingham*. Applicants respectfully submit that the combination of *Dillingham* and *Wolf* fails to teach or suggest these limitations.

As described, the UI server recited in independent claims 8, 20, 37, and 40 stores a UI form definition or identifier corresponding to a client device's platform or capabilities. The same form definition is also cached at the client device and is then populated with content from the UI server, i.e. source data items. The *Dillingham* reference discloses a distinct configuration and operation.

In contrast, the *Dillingham* reference describes a typical Web server transmitting the same HTML and XML instructions to clients, regardless of the clients' platform or capabilities. Col. 3, line 62-Col. 4, line 4. Nowhere does the reference describe storing a form definition or identifier that corresponds to a particular client device as claimed. The reference describes creating a "custom client-side object to cache," col. 9, lines 11-21, but this object is created in response to a user selection, col. 7, lines 32-36, col. 7, line 66-col. 8, line 13, not client platforms or capabilities.

Dillingham also fails to teach or suggest caching a form definition or identifier and populating the same with source data items from the server. Although the reference describes caching the contents of the browser dialog 122, col. 7, lines 33-36, and caching files, folders, and directories, col. 8, lines 6-10, *Dillingham* does not describe caching a UI form definition as claimed. The claimed source data items, i.e. contents, and the UI form definition are distinct and separately stored such that caching the latter, which is claimed, is not taught or suggested by caching the former, which is described by the *Dillingham* reference.

Additionally, *Wolf* fails to teach or suggest modifying *Dillingham* to achieve these limitations. The *Wolf* reference describes a system and method for editing email messages with a full-featured word processor application that is separate from the email client. The reference specifically describes embedded objects that can be edited when opened and displayed in a separate window with the UI provided by the application program that created the object. Col. 9, lines 28-39; *see also* col. 9, lines 40-54 (describing native and foreign frames). However, this description does not teach or suggest a UI server storing and a client caching the same UI form definition corresponding to a client device's platform or capabilities.

(d) *The References Fail to Teach or Suggest Supplementing a Skeletal UI*

Independent claims 8 and 37 further recite supplementing a skeletal UI stored in a first memory location with one or more icons, labels, or menu items, or combinations thereof, stored in a second memory location. The rejection asserts that the *Wolf* reference teaches or suggests these limitations: "Wolf shows the feature at column 9, lines 28-40." Applicants respectfully disagree with this characterization and maintain that the cited references fail to teach or suggest the limitation.

Wolf describes displaying and editing an embedded object in a separate window "with the user interface provided by the application program that created the object. The object...displays the user interface associated with the application program that created the embedded object." Col. 9, lines 28-40. This description clearly discloses a complete user interface, namely that of the application program that created the object. There is no indication in the reference that the "user interface associated with the application program" is "skeletal" or anything less than an entire UI displayed to a user. Merely because such user interface is used to display the content of an embedded object does not suggest that the user interface is "skeletal" as claimed, much less that the user interface is supplemented with icons, labels, or menu items as claimed. Moreover, the reference clearly fails to teach or suggest a first memory location—storing a skeletal UI—and a second memory location—storing icons, labels, or menu items—as recited by claims 8 and 37.

This application now stands in allowable form and reconsideration and allowance is respectfully requested.

Respectfully submitted,

DORSEY & WHITNEY LLP
Customer Number 25763

Date: March 23, 2005

By: Christopher R Hilberg
Christopher R. Hilberg, Reg. No. 48,740
(612) 492-6694